Method and device for coding and decoding structured documents

The invention relates to a method and device for coding and decoding structured documents according to the preamble of claim 1.

XML (=extensible markup language) is a language, which allows a structured description of the content of a document by means of XML schema language definitions. A more detailed description of the XML schema and the structures, data types and content models used therein can be found in the references
- http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/,
- http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/,
- http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

Methods, devices or systems for coding and decoding XML-based documents are known from documents relating to the MPEG-7 standard, in particular ISO/IEC 15938-1 "Multimedia Content Description Interface - Part 1: Systems", Geneva 2002.

Extensions to the method, devices or systems for coding and decoding XML-based documents are known from documents relating to the MPEG-7 standard, from the German application with the official reference 10351897.5. This discloses a method for coding a structured document, in particular an XML-based document, with which a plurality of codes are generated by means of one or more schemas and/or name spaces, with respectively separate codes, which are independent of other schemas and/or name spaces, for the elements defined and/or declared in the schemas and/or name spaces and/or in the groups of schemas and/or name spaces, being allocated for a schema and/or name space and/or a group of schemas and/or name spaces.

These allow efficient coding, even where schemas are not known in full to the encoder and/or decoder. This is achieved in that code tables are separated based on name spaces for the data types, global elements and replacement groups, with a name space referring to a space, in which names of data types (type names) used therein are allocated unique meanings and defined therewith.

Known methods for the binary representation of MPEG-7 and other XML-based descriptions or documents have shortcomings in respect of encoding and decoding complexity, in so far as the XML description or XML document to be coded is based on a number of name spaces. For example, in the documents cited above a method is defined for the binary representation of XML descriptions and XML documents based on schemas and name spaces. (The term "name space" is hereafter used as a synonym for the term "schema").

According to the known method, data types can thereby be bequeathed from other data types. This inheritance relationship allows an instance of a bequeathed type to be used in an XML document instead of an instance of the basic type.

Based on the basic type the type code signals which type an instance is. Based on a basic type in a first name space, the inheritance structure must be analyzed over a number of name spaces during encoding and/or decoding to determine the addressable type names in a second name space. To this end an inheritance tree is established, as described in ISO/IEC 15938-1 "Multimedia Content Description Interface - Part 1: Systems", Geneva 2002.

This assumes that all the name spaces are known and the entire inheritance tree can be established in the memory. The entire inheritance tree comprises the qualifiers of the said types of all the name spaces referenced for the instantiation of an XML description and/or XML document and their inheritance relationship. The described method is therefore very complex.

The object of the invention is now to specify a method and device for coding and decoding structured documents that are simpler than those of the prior art.

This object is achieved based on the method for coding as claimed in the features of the preamble of claim 1, based on the method for coding as claimed in the features of the preamble of claim 2, based on the method for decoding as claimed in the features of the preamble of claim 17 and based on the coding device as claimed in claim 22, the decoding device as claimed in claim 23 and the coding/decoding device as claimed in claim 24, in each instance by their characterizing features.

With the inventive method for coding a structured, in particular XML-based, document, with which a plurality of codes are generated by means of one or more name spaces and allocated for types defined by means of name spaces, a subset of addressable types of one of the name spaces is determined based on inheritance relationships between the name spaces and the name spaces of the basic types of the subset.

The method is advantageously characterized in that only a small number of all the name spaces present has to be stored or loaded to identify the addressable subset. This

significantly reduces the load on resources and also accelerates coding.

Alternatively or additionally, with the method for coding a structured, in particular XML-based, document, for each name space an assignment to further name spaces is carried out such that at least one assignment information item is generated such that at least one inheritance relationship is described between an inheriting name space and bequeathing name spaces. A name space, which contains types, which are directly bequeathed by a basic type from another name space, is hereby referred to as an inheriting name space and a name space, which contains types, which were bequeathed in another name space, is referred to as a bequeathing name space.

The assignment information provided by this development allows a structured organization of inheritance information, such that only a part of the entire inheritance tree is required to identify the subset. This development thus results in further savings and load reduction in respect of resources as well as acceleration.

The assignment information of the inheriting name space is preferably formed from a list of codes of the basic types of header types of the inheriting name space, with header types being types, which originate directly from a basic type of the bequeathing name space, and with the basic types also being formed by header types, from which further header types result.

The addressable subset is preferably determined based on an initial basic type of the basic types of the bequeathing name space, with header types being identified in the inheriting

name space by the assignment information to identify the subset, generally based on the initial basic type for determining the subset, said header types originating from a basic type from the bequeathing name space, with which the initial basic type is a basic type in the bequeathing name space.

Alternatively or additionally, with the method for coding a structured, in particular XML-based, document for at least one name space, the assignment information assigned to the inheriting name spaces is stored together with the respective name space in a first device carrying out the coding and/or decoding.

In one development, the assignment information assigned to the inheriting name spaces is generated in a second device and transmitted together with the respective name space in a first device carrying out the coding and/or decoding.
The method described here is advantageous, as now only the name space of the basic type, the name space of the data type to be addressed and the inheritance relationship BT have to be known and/or loaded to determine the addressable data types.

One important advantage of the inventive method is that it allows efficient determination of the addressable type names, without having to establish the entire inheritance tree. Furthermore this can also be done without knowledge of all the schemas or name spaces.

A further advantage is that the search for the addressed data type can be carried out with fewer comparison operations compared with the search in the entire inheritance tree.

In one embodiment the inheritance information BT of a name space NS comprises a list of type codes TC^{LBT} of the basic types LBT for each header type HT of the name space NS.

In a further embodiment the type codes are allocated according to the following method to this end:

To code a structured document, a plurality of codes are generated by means of one or more schemas and/or name spaces. Separate codes, which are independent of other schemas and/or name spaces for the elements defined and/or declared in the schemas and/or name spaces and/or in the groups of schemas and/or name spaces, are thereby allocated for a schema and/or a name space and/or for a group of schemas and/or name spaces

In a development of this, codes are allocated separately in schemas and/or name spaces. The method described here is advantageous, as schemas and/or name spaces can now be loaded as required even during the transmission of documents and existing code tables for other name spaces do not change as a result and therefore do not have to be recreated. A further advantage is that the separate codes for instances where a large number of name spaces are imported require fewer bits for addressing purposes than if all name spaces are combined, as in ISO/IEC 15938-1 "Multimedia Content Description Interface - Part 1: Systems", Geneva 2002. The separate codes for the other name spaces can be coded with fewer bits even in instances where a very large name space is imported.

In one preferred variant of the invention the separate codes are divided into address areas, it being possible to identify the schema and/or name space or group of schemas and/or name spaces via the address areas.

In one preferred embodiment of the inventive coding method, the separate codes each comprise a local code relating to the schema and/or name space and/or relating to the group of schemas and/or name spaces and an identification code, which identifies the schema and/or name space and/or the group of schemas and/or name spaces. A local code here is a code, which is unique within the schema or name space identified by the identification code.

Separate codes are preferably allocated for global elements and/or substitution groups and/or data types. A precise definition of global elements, substitution groups and data types can be found in the XML schema definitions, which are described in detail in the documents - http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

For data types type codes, as described in the document ISO/IEC 15938-1 "Multimedia Content Description Interface - Part 1: Systems", Geneva 2002, in a preferred embodiment separate codes are generated such that within the inheritance tree of a name space the data type adjacent to a first data type in the same name space is at a code interval from the first data type, said interval corresponding to the number of data types derived from the first data type in this name space. A data type is adjacent to a first data type, when the data type is derived from the same basic data type as the first data type and the smallest type code has been assigned to the data type, out of all the data types derived from this basic data type, said type code being greater than the type code of the first data type. With this embodiment the codes

for the data types type codes are allocated such within the - possibly disjoint - inheritance tree, that an advantageous adjacency relationship results and is maintained in a given name space, even if sub-trees containing types derived from other name spaces occur in this name space.

In a particularly preferred embodiment of the inventive method, the separate codes are allocated within a given name space according to a method comprising the following steps:

- in a first step all data types of a name space, which were bequeathed from data types of other name spaces, are sorted in a list in the sequence of global type codes of the respective basic data types as defined in the MPEG-7 standard, the basic data types being the data types in other name spaces, from which the sorted data types were bequeathed;
- in a second step those data types of a name space, which were bequeathed from a specific basic data type of a specific other name space, are sorted lexicographically in each instance;
- in a third step all the data types of a name space, which were not bequeathed from a data type of another name space, are sorted according to the sequence defined in the MPEG-7 standard into the existing list of data types;
- in a fourth step the separate codes are allocated in list sequence to the data types of the name space.

The advantage of this embodiment is that the addressed data type, in particular a type code, can be quickly found and thus decoded. According to the rules in ISO/IEC 15938-1 "Multimedia Content Description Interface - Part 1: Systems", Geneva 2002, a type code addresses a derived type relative to a basic type. The basic type thus defines a sub-tree, in which all

addressable data types are present. If a number of name spaces are contained in the sub-tree, the advantageous adjacency relationship, as achieved with the above embodiment of the invention, means that an addressed data type can be found quickly in the name space, as it can be established by comparing a searched for data type with two adjacent data types in the sorted inheritance tree, whether the searched for data type is in the sub-tree of the data type of the two adjacent data types with the smaller binary code. This can significantly reduce the time and effort required for the search. A further advantage of this adjacency relationship is that when coding the type codes according to ISO/IEC 15938-1 "Multimedia Content Description Interface - Part 1: Systems", Geneva 2002, a decoder can calculate the code word length, which is determined directly by the number of derived data types, directly from the code interval of the adjacent data types.

In a further embodiment, the local type codes are allocated according to the method described above, with the type code TC^{LBT} being formed from the name space ID and local type code according to the method described above in one development. In a further embodiment the local type codes are allocated according to the method described above and only basic types of the first name space are considered, the local type code of which

a) is greater than the local type code of the initial basic type OBT and

b) is smaller than the smallest, next largest local type code of a type adjacent to the initial basic type OBT.

In one preferred embodiment of the inventive method, the

inheritance relationships BT between name spaces are stored and/or transmitted with a schema and/or name space.

In addition to the inventive coding method described above, the invention also relates to a decoding method, with which a structured document, in particular an XML-based document, is decoded, with the method being configured such that a document coded with the inventive coding method is decoded.

In one preferred embodiment of the inventive decoding method, the code length of the separate code for the binary type code is determined from the number of derived data items to decode a binary type code - the generation of which is described above. Furthermore, in a preferred embodiment for decoding a specific type code of the sub-tree of the inheritance tree of the name space, in which the specific type code is located, [lacuna] is preferably determined from the code intervals between adjacent data types.

In one development, to determine the basic types originating from an initial basic type [lacuna] is determined from the code interval between adjacent data types.

In a further alternative or addition, [lacuna] is determined to establish the number of types in the subset based on the header types using the code intervals between adjacent header types.

In addition to the method described above, the invention also relates to a coding device and a decoding device to implement the inventive coding and/or decoding method. The invention also comprises a coding and decoding device, with which the

inventive coding method and the inventive decoding method can be implemented.

Exemplary embodiments of the invention are described in more detail below with reference to the accompanying drawings, in which:

Figure 1 shows a basic diagram of a coding and decoding system in which the inventive method is brought to bear.

Figure 2 shows a diagram of an exemplary XML schema definition, in which data types from other name spaces are also imported and derived.

Figure 3 shows a diagram of an inheritance tree of data types, including the assignment of local codes to types occurring in the name spaces.

Figure 4 shows a diagram of an inheritance tree of data types, which extends over a number of name spaces.

Figure 5 shows a diagram of an inheritance tree including inheritance information between name spaces.

Figure 1 shows an example of a coding and decoding system, in which the inventive method is used, with an encoder ENC and a decoder DEC, with which XML documents are coded and/or decoded. Both the encoder and the decoder have what is known as an XML schema S, in which the elements and types of the XML document used for communication are declared and defined. Code tables CT are generated in the encoder and decoder from the schema S via corresponding schema compilations SC. When the XML document DOC is coded, binary codes are assigned to the

content of the XML document via the code tables. This generates a binary representation BDOC of the document DOC, which can be decoded again in the decoder with the aid of the code table CT. A number of schemas can be used here, in particular schemas can be deployed which are based on a basic schema and are derived from a further schema.

Figure 2 shows an exemplary extract from an XML schema definition. Such XML schema definitions are known to the person skilled in the art, so there is no need too look further at the exact content of the extract in Figure 2. The extract contains two schema definitions. On the one hand a schema A is defined in the upper part, as shown by a curly bracket, and on the other hand a schema X is defined in the lower part, similarly shown by a curly bracket. The schema X in turn uses data types, which have been imported from the schema A.

Figure 3 shows a graphic diagram of the inheritance relationships between a first name space NS1 and a second name space NS2 and their data types in the form of a section of a tree structure. As can be seen from the backward pointing, non-dashed arrow in the figure, there is an inheritance relationship between the second name space NS2 and the first name space NS1. Each node in the inheritance tree represents defined, named data type in the schema definition. With the method described in the German application with the official reference 10351897.5, local codes are allocated respectively for the name spaces NS1 .. NS2, as specified in Figure 3 by the numbers to the left of the nodes. These so-called local type codes address all types in a name space uniquely. With the inventive signaling of a data type based on an initial basic type OBT in the first name space NS1, the set of

addressable types in the second name space NS2 is a subset TM - shown by the dashed border - of all types in the name space NS2. Only a few type codes are therefore used, as specified in Figure 3 by the numbers to the right of the nodes and optionally only require a shorter binary representation.

Figure 4 shows four name spaces NS1 .. NS4, between which indirect inheritance relationships also exist; i.e. inheritance relationships, with which at least one further name space from the name spaces NS1 .. NS4 exists between a derived type and a basic type.

According to the invention a set TM - shown by the dashed border in the diagram - of addressable types can now be defined based on the highlighted initial basic type OBT shown in the first name space NS1, by considering the inheritance relationships of all name spaces NS1 .. NS4.

Figure 5 shows the inheritance information BT1 .. BT3 structured according to one variant of the inventive method for the fourth name space NS4.

It can be seen that the structuring is such that for every data type, which is bequeathed directly from a data type from another name space (the respective direct inheritance is shown in each instance by a non-dashed backward pointing arrow), the inheritance information BT1 .. BT3 is stored and/or transmitted from a first device to a second device, with the inheritance information BT1 .. BT3 of a data type comprising a qualifier of the respective name space NAMESPACE_ID and the local type code of the basic types LBT, LBT' in the respective bequeathing name space NS1, NS3 and NS4 according to the variant shown.

Based on this structured inheritance information BT1 .. BT3 the essence of the inventive method becomes clear in that instead of the entire inheritance tree, which results from the union of sets for the inheritance conditions of all name spaces, only the name spaces of initial basic types OBT, the types to be addressed and the inheritance relationship BT1 .. BT3 between the name space (inheriting name space) containing the addressable types and the name space (bequeathing name space) containing the respective initial basic type OBT, are stored and used to determine the addressable data types TM.

The inheritance information (relationship) BT1 .. BT3 thus essentially identifies basic types of header types of a name space, header types referring to the data types based directly on a basic type from a bequeathing name space, with those data types LBT of a bequeathing name space NS1, NS3 and NS4, which

a) are the direct basic type (LBT) of a header type (HT) or
b) are the direct basic type (LBT') of a header type (HT') of any further name space in the inheritance hierarchy - in the exemplary embodiment the third name space NS3 - with the header type HT' being the direct or indirect basic type of the header type HT in the name space of the derived type, being basic types.

Based on this inventive structuring, with the inventive method the addressable data types TM are now identified in a second name space NS2 based on an initial basic type OBT in a first name space NS1, in that
a) header types HT are identified in the second name space, for which types LBT and LBT' from the first name space are

stored as inheritance information BT1 .. BT3 and
b) the type OBT is a basic type of the types LBT and LBT'.

A set of addressable types can thus simply be determined together with the inheritance tree of the inheriting name space based on a data type of the bequeathing name space NS1 to be established as a basic type, without having to have knowledge of all name spaces imported in the inheriting name space. This saves a tremendous amount of time and effort, minimizing computing and memory capacity requirements and accelerating coding and decoding.